Universidad Nacional del Litoral
Algoritmos y Estructuras de Datos

Cátedra e información de la materia. Contacto

Carreras: Ingeniería en Informática, Analista en Informática Aplicada
Extensión: Cuatrimestral
Carga horaria: 90 hs de clase (30 hs de teoría / 60 hs de práctica)
Docentes:

Dr. Mario A. Storti (mario.storti@gmail.com)
Dr. Rodrigo R. Paz (rodrigo.r.paz@gmail.com)
Dr. Lisandro D. Dalcin (dalcinl@gmail.com)

Ayudantes de cátedra:

Diego Galizzi (dgalizzi@gmail.com)
Juan D. Prigioni (jdprigioni@gmail.com)

Programa analítico

Ver “Programa analítico” en la plataforma

1. **Diseño y análisis de algoritmos.** De los problemas a los programas. Tipos de datos abstractos (TDA). Tipos de datos, estructura de datos y tipos de datos abstractos.
2. **Tipos de datos abstractos fundamentales.** El tipo de dato abstracto ``Lista''. Realización de listas. Pilas. Colas. Correspondencias.
3. **Arboles.** Terminología fundamental. El TDA ``árbol''. Realización de árboles. Arboles binarios.
4. **Operaciones básicas con conjuntos.** Introducción a los conjuntos (sets). Un TDA con UNION, INTERSECCION y DIFERENCIA. Realización de conjuntos mediante vectores de bits. Realización de conjuntos mediante listas enlazadas. El diccionario. Realizaciones sencillas de diccionarios. La estructura de datos tabla de dispersión. Estimación de la eficiencia de las funciones de dispersión. Realización del TDA CORRESPONDENCIA. Colas de prioridad. Realización de colas de prioridad. Algunas estructuras complejas de conjuntos.
5. **Métodos Avanzados de Representación de Conjuntos.** Arboles binarios de búsqueda. Tries.
6. **Clasificación.** El modelo de clasificación interna. Algunos esquemas simples de clasificación. Clasificación rápida (Quicksort). Clasificación por montículos (Heapsort).

7. **Técnicas de análisis de algoritmos.** Eficiencia de los algoritmos. Tiempo de ejecución de un programa. Cálculo del tiempo de ejecución de un programa. Buenas prácticas de programación. Análisis de programas recursivos.
8. **Técnicas de diseño de algoritmos.** Algoritmos dividir para vencer. Programación dinámica. Algoritmos ávidos (greedy). Método de retroceso (Backtracking). Algoritmos de búsqueda local.

Modalidad de dictado

Se dictaran clases

- Teóricas (2.5 hs semanales)
- Prácticas (4.5 hs semanales)
- De consulta (2 hs semanales)

Evaluación

La evaluación se realiza mediante **tres exámenes parciales**. La regularidad se logra aprobando todos los parciales. Para ello deben contar con una **calificación mínima de 40%** y obtener un **porcentaje mínimo en cada sección del parcial**. Este porcentaje mínimo puede variar, pero típicamente es de **60% en preguntas, 40% en programación, 25% en operativos, y 50% en clases**.

Lograrán **promoción directa** aquellos **alumnos regulares** que obtengan un promedio en los parciales de al menos **70 sobre 100 puntos**. Se calcula una **Nota Final** igual al promedio de las notas de los Exámenes Parciales.

A parte de cumplir los anteriores requisitos el alumno deberá presentar **por lo menos el 70%** de los ejercicios propuestos en las **guías de trabajo práctico**, tanto para **regularizar** como para **promocionar**.

Se realizará un **Examen Recuperatorio**. Las reglas del mismo son:
El recuperatorio se toma normalmente en la semana 16. Las reglas para ver qué se puede rendir son así. Si dividimos los 3 parciales en las 4 secciones correspondientes, tenemos un total de 12 secciones, a saber CLASES1, PROG1, OPER1, PREG1 en el Parcial 1, CLASES2, PROG2, OPER2, PREG2 en el Parcial 2 y CLASES3, PROG3, OPER3, PREG3 en el Parcial 3. Se pueden rendir **HASTA 6** de esas secciones en forma completamente independiente, es decir se pueden rendir por ejemplo: CLASES1, PROG1, CLASES2, OPER2, PROG3 y PREG3.

Se puede rendir para **regularizar** (por no haber llegado a algún mínimo, o por promedio) **O** para **promocionar**, **O** para **mejorar la nota**.

La nota obtenida reemplaza a la correspondiente anterior, es decir si se rinde OPER2 entonces esa nueva nota reemplaza a la anterior en OPER2. Como consecuencia, si algún alumno esta impedido de regularizar o promocionar por no tener aprobada alguna de las secciones, entonces **necesariamente debe recuperar esas secciones**.

Si es para **regularizar** entonces no hay restricción con respecto al número de secciones, es decir se pueden rendir las 12 secciones.

Sólo se reemplaza la nota de un punto **si la obtenida en el recuperatorio es mejor**.

En cada sección que se recupere se exige un **mínimo de un 60%** de puntaje. Si no se llega a esta mínimo el puntaje en esa sección se considera 0, y por lo tanto no reemplaza al previo.

Aquellos alumnos que no logren promoción directa, deberán rendir un examen final.

Cronograma tentativo

Inicio de clases: Semana 1.
Primer parcial: Semana 5.
Segundo parcial: Semana 11.
Tercer parcial: Semana 15.
Recuperatorio: Semana 16.

Enlaces de interés

- http://www.cs.princeton.edu/~rs/shell/animate.html: Clasificación animada interactiva (se puede cambiar la secuencia de algoritmos que trabajan con distintos incrementos), se ve muy bien el último paso en clasificación shellsort
- http://oopweb.com/Algorithms/Documents/PLDS210/VolumeFrames.html: OOP, algoritmos y estructura de datos. Parece una buena referencia online.
- http://www.brpreiss.com/books/opus4/: También, una muy buena referencia online sobre OOP, algoritmia y estructuras.
- http://www-cg-hci.informatik.uni-oldenburg.de/~da/peters/Kalvin/Doku-SN.htm: Algoritmos de clasificación y sus fuentes. Animaciones.